KGP RISC V

Group 52
Mradul Agrawal 20CS30034
Swapnil Agrawal 20CS30054

Instructions

1. Shift Instructions

Instruction	Name	Meaning
shll rs, rh	Shift Left Logical	rs ← (rs) left-shifted by sh
shrl rs, rh	Shift Right Logical	rs ← (rs) right-shifted by sh
shllv rs, rt	Shift Left Logical Variable	rs ← (rs) left-shifted by (rt)
shrlv rs, rt	Shift Right Logical Variable	rs ← (rs) right-shifted by (rt)
shra rs, rh	Shift Right Arithmetic	rs ← (rs) arithmetic right-shifted by sh
shrav rs, rt	Shift Right Arithmetic Variable	rs ← (rs) right-shifted by (rt)

2. Arithmetic Instructions

Instruction	Name	Meaning
add rs, rt	Add	rs ← (rs) + (rt)
comp rs, rt	Comp	rs ← 2’s Complement (rt)
addi rs, imm	Add Immediate	rs ← (rs) + imm
compi rs, imm	Comp Immediate	rs ← 2’s Complement (imm)

3. Logic Instructions

Instruction	Name	Meaning
and rs, rt	And	rs ← (rs) ∧ (rt)
xor rs, rt	Xor	rs ← (rs) ⊕ (rt)

4. Memory Instructions

Instruction	Name	Meaning
lw rt, imm(rs)	Load Word	rt ← mem[(rs) + imm]
sw rt, imm(rs)	Store Word	mem[(rs) + imm] ← (rt)

5. Branch Instructions

Instruction	Name	Meaning
b L	Unconditional Branch	goto L
br rs	Branch Register	goto (rs)
bltz rs, L	Branch on less than zero	if(rs) < 0 then goto L
bz rs, L	Branch on flag zero	if (rs) = 0 then goto L
bnz rs, L	Branch on flag not zero	if(rs)≠ 0 then goto L
bl L	Branch and link	goto L; 31 ← (PC)+4
bcy L	Branch on Carry	goto L if Carry = 1
bncy L	Branch on no Carry	goto L if Carry = 0

6. Complex Instructions

Instruction	Name	Meaning
diff rs, rt	Diff	rs ← the LSB bit at which rs and rt differrs ← the LSB bit at which rs and rt differ

Instruction Format and Encoding

1. Shift Instructions

Field Size	6 Bit	5 Bit	5 Bit	5 Bit	6 Bit	5 Bit
Format	Op Code	rs	rt	Shift Amount	Fun Code	Extra

For shll, shrl, shllv, shrlv, shra, shrav

2. Arithmetic Instructions

Field Size	6 Bit	5 Bit	5 Bit	5 Bit	6 Bit	5 Bit
Format	Op Code	rs	rt	Exra	Fun Code	Extra

For add, comp

3. Logic Instructions

Field Size	6 Bit	5 Bit	5 Bit	5 Bit	6 Bit	5 Bit
Format	Op Code	rs	rt	Exra	Fun Code	Extra

For and, xor

4. Complex Instructions

Field Size	6 Bit	5 Bit	5 Bit	5 Bit	6 Bit	5 Bit
Format	Op Code	rs	rt	Exra	Fun Code	Extra

For diff

5. Arithmetic Immediate Instructions

Field Size	6 Bit	5 Bit	5 Bit	16 Bit
Format	Op Code	rs	Extra	Immediate Value

For addi, compi

6. Memory Instructions

Field Size	6 Bit	5 Bit	5 Bit	16 Bit
Format	Op Code	rs	rt	Address Value

For sw, lw

7. Branch Instructions Type 1

Field Size	6 Bit	26 Bit
Format	Op Code	Address Value

For b, bl, bcy, bncy

8. Branch Instructions Type 2

Field Size	6 Bit	5 Bit	5 Bit	21 Bit
Format	Op Code	rs	Extra	Address Value

For bltz, bz, bnz

9. Branch Instructions Type 3

Field Size	6 Bit	5 Bit	21 Bit
Format	Op Code	rs	Extra

For br

Instruction Encoding

1. Shift Instructions

Instruction	OP Code	Fun Code
shll	000000	001100
shllv	000000	001000
shrl	000000	001110
shrlv	000000	001010
shra	000000	001111
shrav	000000	001011

2. Arithmetic Instructions

Instruction	OP Code	Fun Code
add	000000	000001
comp	000000	000101

3. Logical Instructions

Instruction	OP Code	Fun Code
and	000000	000010
xor	000000	000011

4. Complex Instructions

Instruction	OP Code	Fun Code
diff	000011	000000

5. Arithmetic Immediate Instructions

Instruction	OP Code
addi	001000
compi	001001

6. Memory Instructions

Instruction	OP Code
lw	010000

sw	011000

7. Branch Instructions

Instruction	**OP Code**
b	101000
br	100000
bcy	101001
bncy	101010
bltz	110000
bz	110001
bnz	110010
bl	101011

Control Signals

1. Shift Module

Instruction	**Type**	**Direction**
shll shllv	0	0
shrl shrlv	0	1
shra shrav	1	X

For shifter:
ALUOp1 = Type
ALUOp0 = Direction

2. ALU Module

Instruction	**ALUOp3**	**ALUOp2**	**ALUOp1**	**ALUOp0**
shll	0	0	0	0
shllv	0	1	0	0
shrl	0	0	0	1
shrlv	0	1	0	1
shra	0	0	1	0
shrav	0	0	1	1

and	1	0	0	0
xor	1	0	0	1
add/addi/lw/sw	1	0	1	0
comp/compi	1	1	1	1
diff	1	0	1	1
br	1	1	1	0

3. Control Module

a. Shift Instructions

Instruction	**ALUSel**	**RegW**	**MemR**	**MemW**	**MemReg**	**PCSel**
shll	1	1	0	0	00	0
shllv	0	1	0	0	00	0
shrl	1	1	0	0	00	0
shrlv	0	1	0	0	00	0
shra	1	1	0	0	00	0
shrav	0	1	0	0	00	0

b. Arithmetic Instructions

Instruction	**ALUSel**	**RegW**	**MemR**	**MemW**	**MemReg**	**PCSel**
add	0	1	0	0	00	0
comp	0	1	0	0	00	0

c. Logic Instructions

Instruction	**ALUSel**	**RegW**	**MemR**	**MemW**	**MemReg**	**PCSel**
and	0	1	0	0	00	0
xor	0	1	0	0	00	0

d. Immediate Instructions

Instruction	**ALUSel**	**RegW**	**MemR**	**MemW**	**MemReg**	**PCSel**
addi	1	1	0	0	00	0
compi	1	1	0	0	00	0

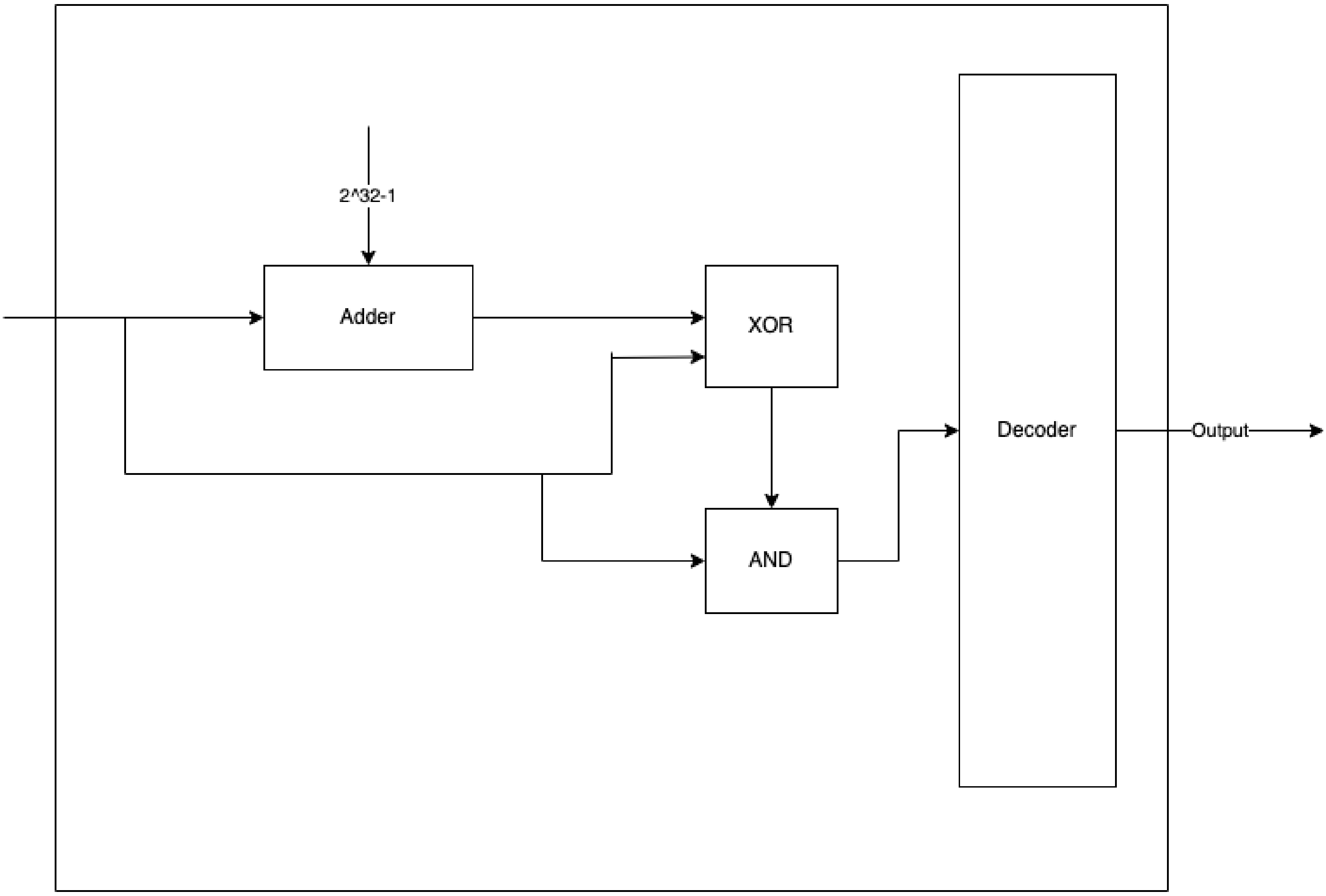
e. Memory Instructions

Instruction	ALUSel	RegW	MemR	MemW	MemReg	PCSel
lw	1	1	1	0	01	0
sw	1	0	0	1	00	0

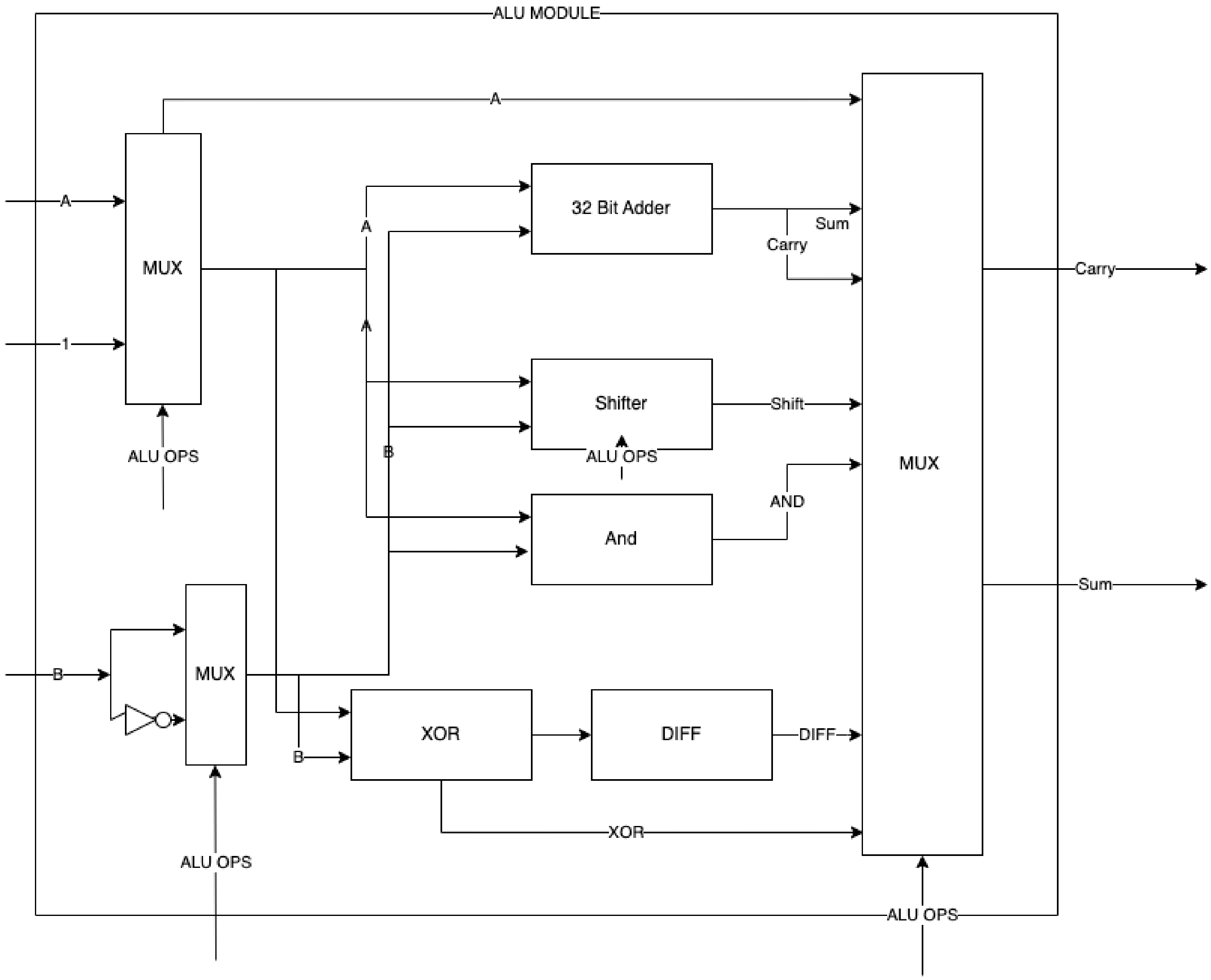
f. Branch Instructions

Instruction	ALUSel	RegW	MemR	MemW	MemReg	PCSel
b	X	0	0	0	00	1
br	X	0	0	0	00	1
bltz	X	0	0	0	00	1
bz	X	0	0	0	00	1
bnz	X	0	0	0	00	1
bl	X	0	0	0	10	1
bcy	X	0	0	0	00	1
bncy	X	0	0	0	00	1

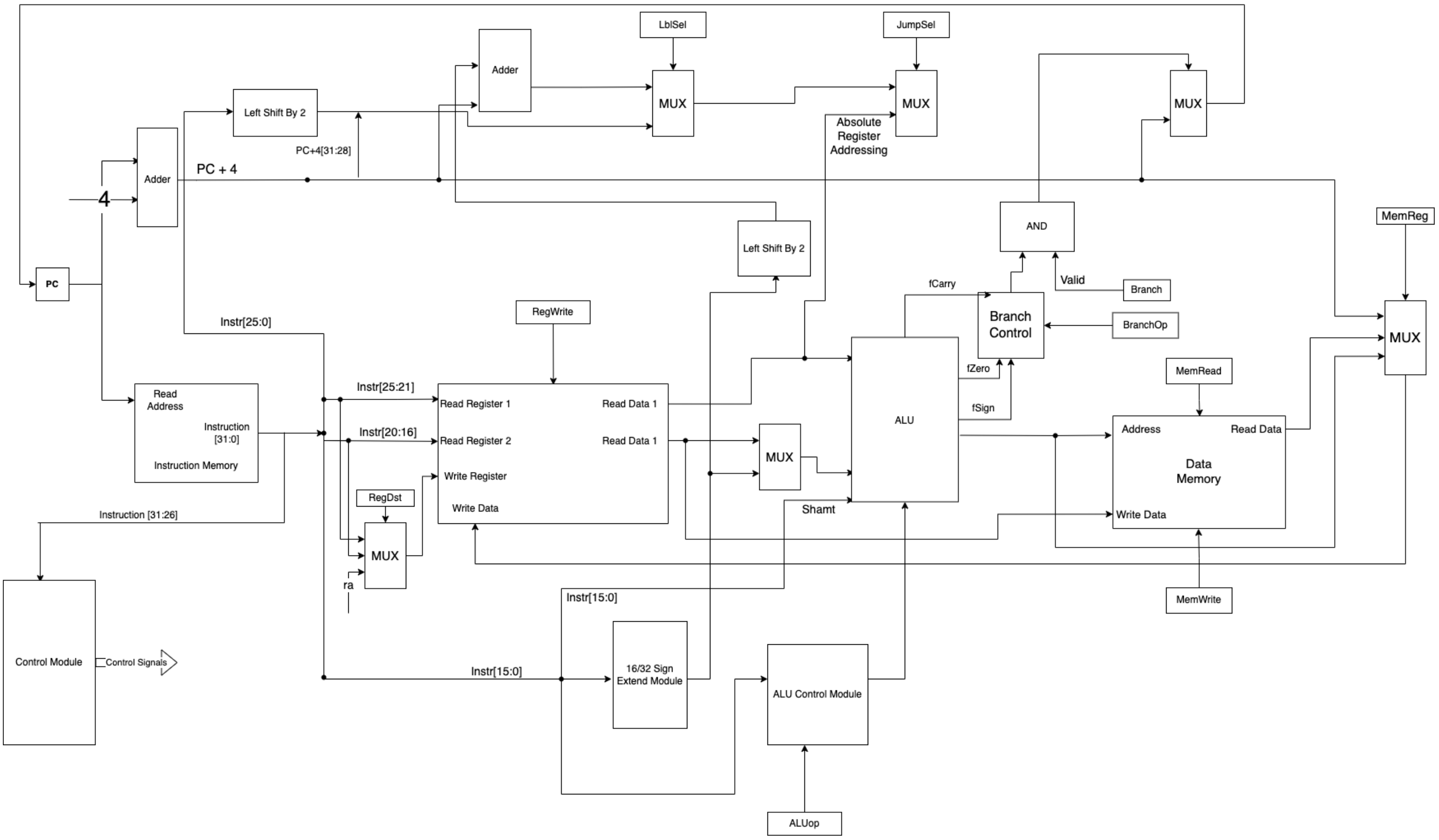
DIFF Module Diagram



ALU Diagram

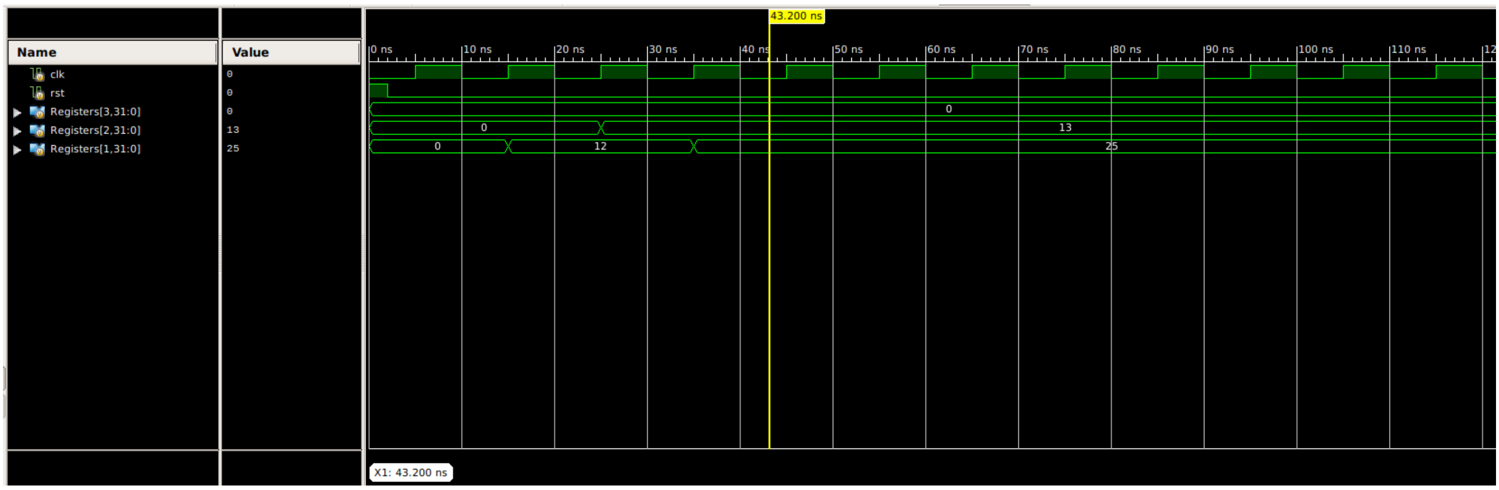


Data Path Diagram



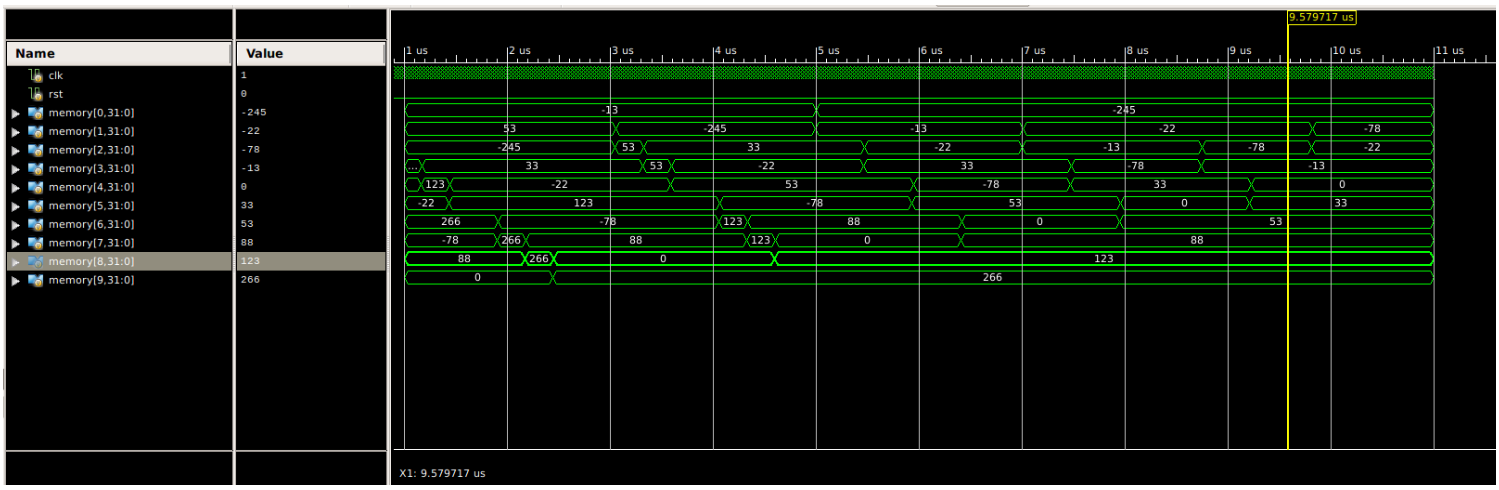
Target Programs

1. Basic Add

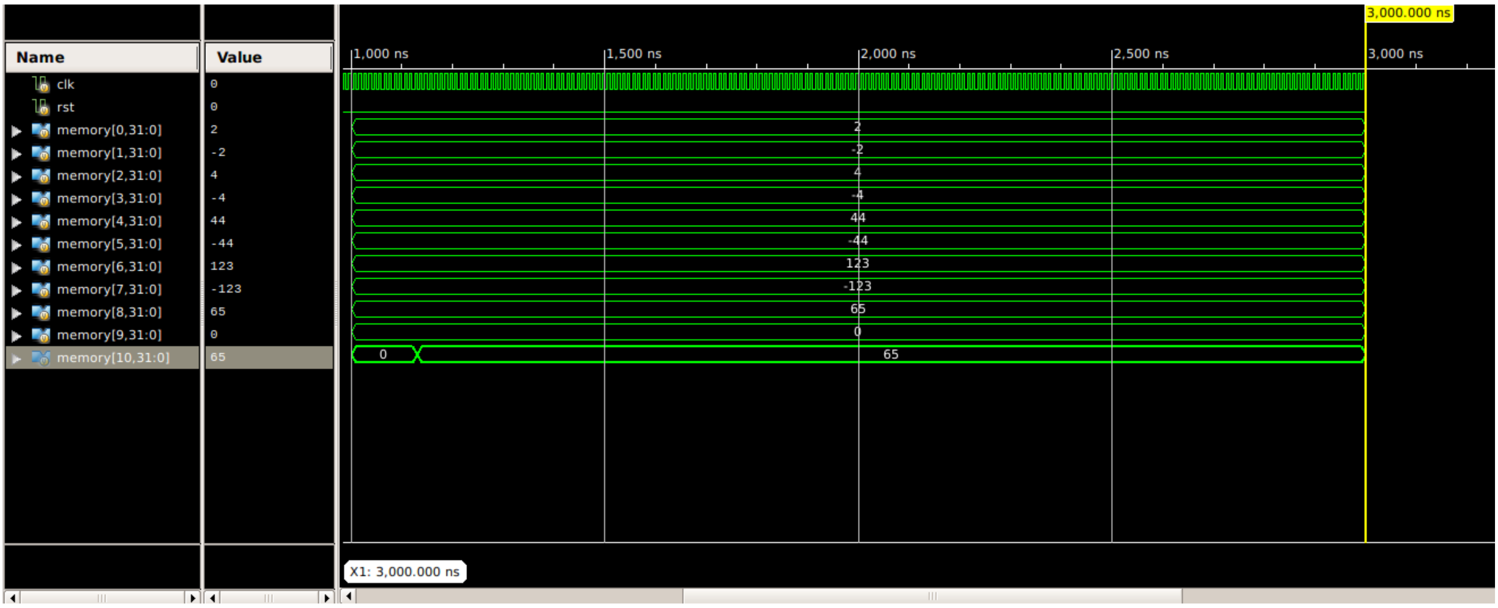


Name	Value
clk	0
rst	0
Registers[3,31:0]	0
Registers[2,31:0]	13
Registers[1,31:0]	25

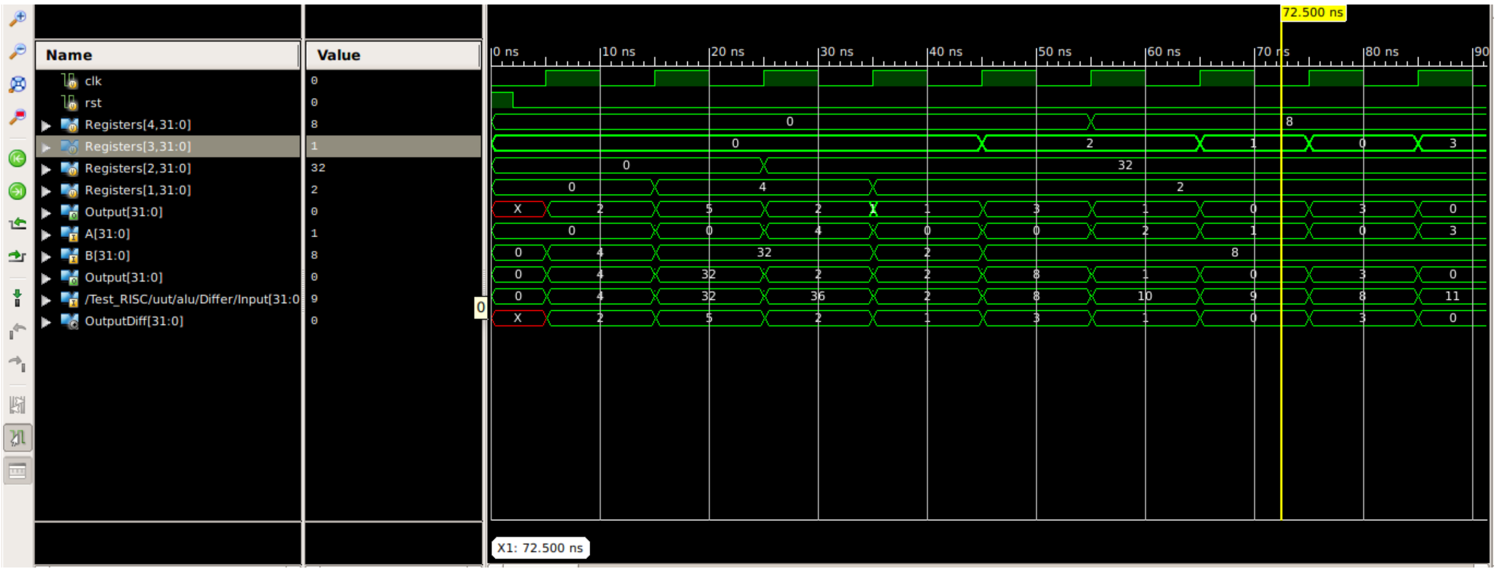
2. Bubble Sort



3. Array Sum



4. DIFF Test


Name
Value
clk 0
rst 0
Registers[4,31:0] 8
Registers[3,31:0] 1
Registers[2,31:0] 32
Registers[1,31:0] 2
Output[31:0] 0
A[31:0] 1
B[31:0] 8
Output[31:0] 0
/Test_RISC/uut/alu/Differ/Input[31:0] 9
OutputDiff[31:0] 0
72.500 ns
X1: 72.500 ns